

# Estructuras de Datos y Algoritmos

## Grados en Ingeniería Informática, de Computadores y del Software

Examen Final, 12 de junio de 2013.

### 1. (3 puntos)

Se tiene un vector de enteros  $a[0..n-1]$ , que puede ser vacío, cuyo propósito es representar los coeficientes de un polinomio en una variable:  $a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$ . Se pide especificar, y derivar, o diseñar y verificar una función iterativa que, dado un vector de coeficientes y un valor de  $x$ , devuelva la evaluación del polinomio para ese valor. El coste ha de ser **lineal** y se ha de justificar su cálculo. La evaluación ha de seguir **obligatoriamente** la secuencia de cálculos impuesta por la llamada Regla de Horner:

$$p(x) = a_0 + a_1(\dots a_{n-3} + (a_{n-2} + a_{n-1}x)x \dots)x$$

Si el polinomio es vacío, se entenderá que su evaluación para cualquier  $x$  da cero.

### 2. (4 puntos)

Desarrolla MultaMatic, el nuevo sistema de gestión de multas de tráfico por exceso de velocidad. La red de carreteras contiene tramos vigilados en los que se coloca una cámara al principio del tramo y otra al final. Cada vez que un coche pasa frente a una cámara, se toma una foto de su matrícula y se apunta el momento en que pasó; si el tiempo transcurrido entre la foto del comienzo y la del final es demasiado breve, se le pone una multa. Para simplificar, asumiremos que los tramos no comparten cámaras ni se solapan entre sí. Las operaciones públicas del TAD son:

- *insertaTramo*: añade un nuevo tramo al sistema. Recibe un identificador de tramo, los identificadores de sus cámaras inicial y final, y el número mínimo de segundos que deben transcurrir entre las fotos de comienzo y final para *no* recibir multa. Si el tramo ya existía debe generar un error.
- *fotoEntrada*: se invoca cada vez que un coche entra en un tramo vigilado. Recibe el identificador de la cámara, la matrícula del coche, y el instante actual (en segundos desde el 1 de enero de 1970).
- *fotoSalida*: se invoca cada vez que un coche sale de un tramo vigilado. Recibe el identificador de la cámara, la matrícula del coche, y el instante actual. Si el coche ha ido demasiado rápido en el tramo, se le multará.
- *multasPorMatricula*: devuelve el número de multas asociadas a una matrícula.
- *multasPorTramos*: devuelve una lista con las matrículas de los coches multados en un determinado tramo. Si un coche ha sido multado varias veces, su matrícula aparecerá varias veces en la lista. Si el tramo no existe debe generar un error.

Se pide: prototipos de las operaciones públicas, representación eficiente del TAD (basada en tipos vistos durante el curso), coste de cada operación e implementación en C++ de todas las operaciones.

### 3. (3 puntos)

Dada una lista de números  $a_1, a_2, \dots, a_n$  se dice que dos números producen una inversión si  $a_i > a_j$  con  $i < j$ . Se pide un algoritmo que calcule el número de inversiones que existen en una lista de números. Se debe utilizar el método **Divide y Vencerás**. Se valorará la eficiencia del algoritmo implementado. **Nota:** Se puede conseguir un coste del orden de  $O(n \log n)$ .

Calcular el coste del algoritmo implementado planteando la recurrencia y resolviéndola.